Fig 1
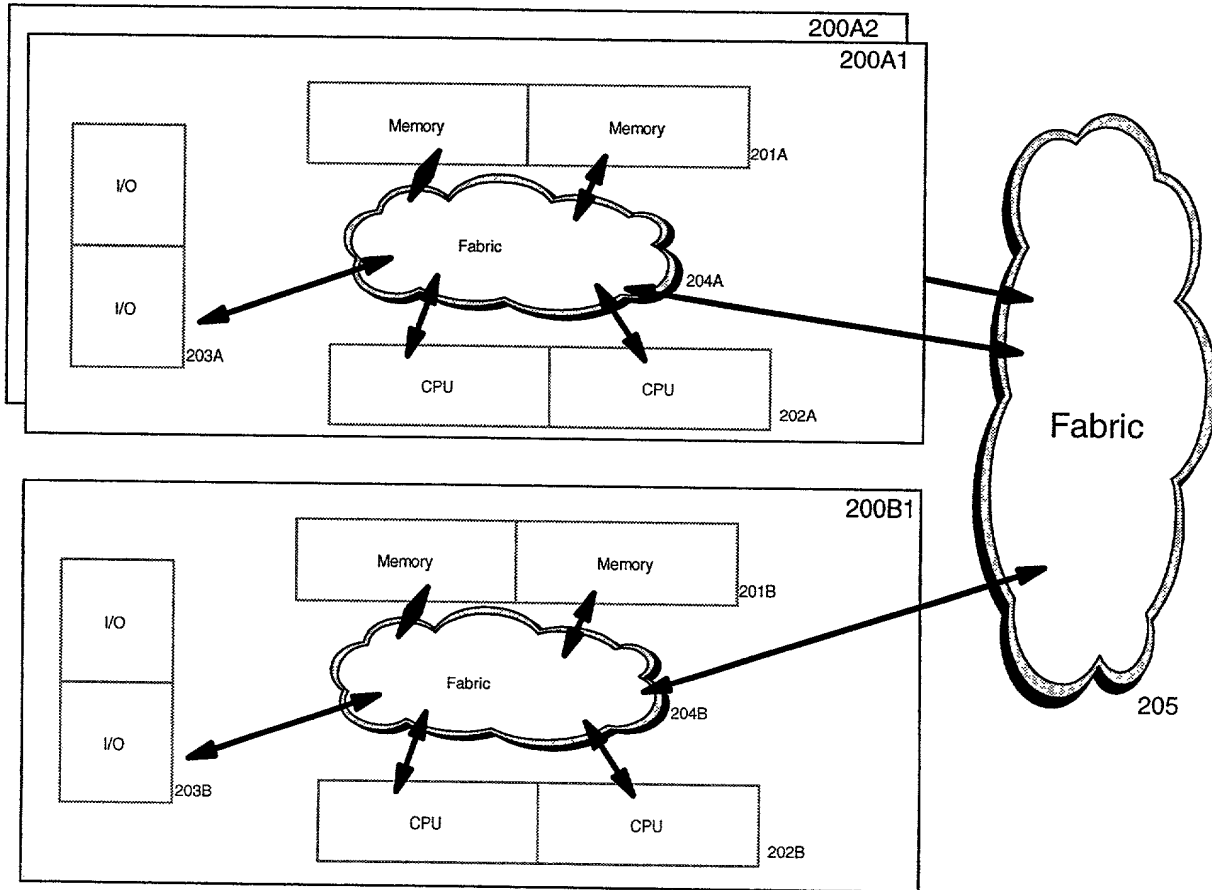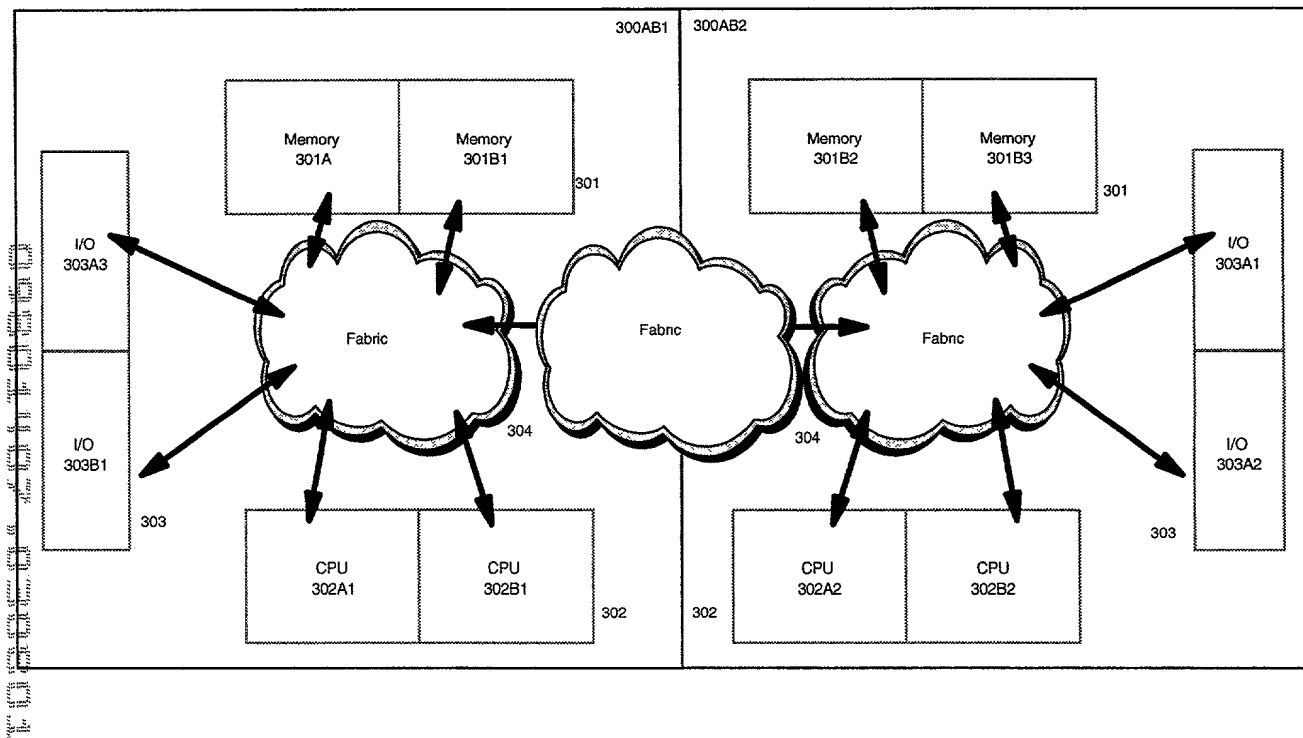
Fig 2

Fig 3

Virtualization allows sharing of CPUs and I/O elements by multiple partitions

400
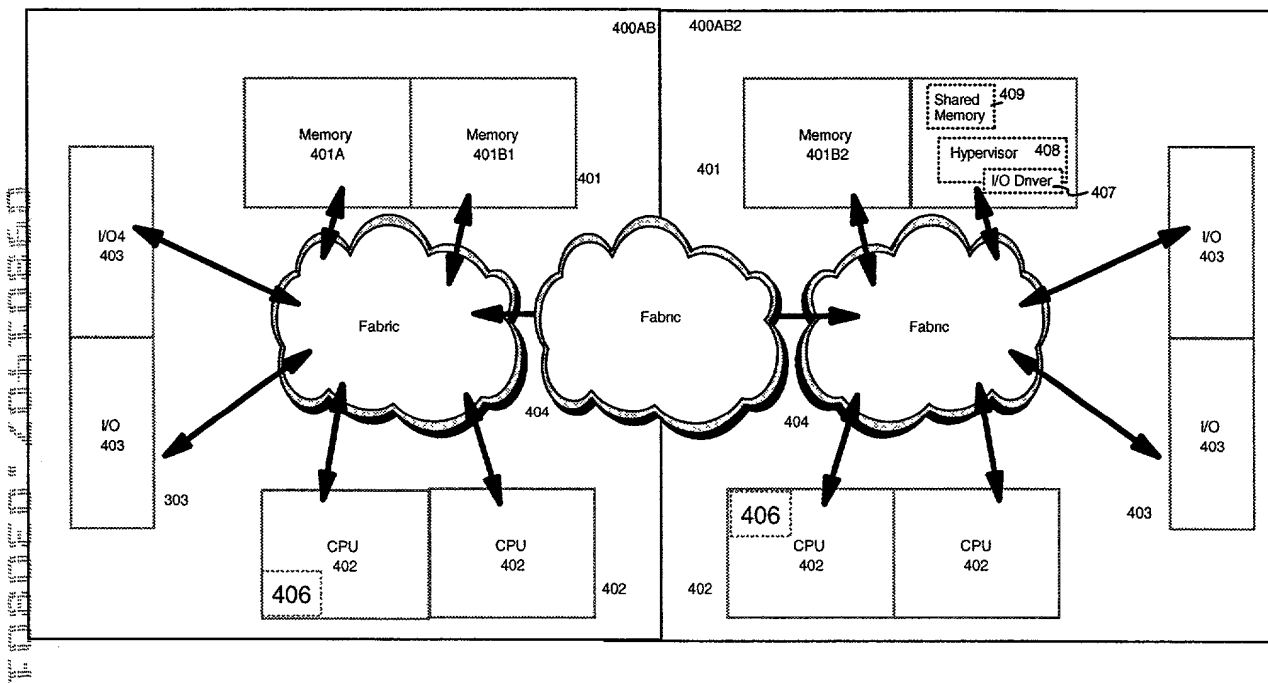

Fig 4

Fig 5

Fig 6

Fig 7A

Process A
701

Memory A
702

714

715

Process A
703

Memory A
704

Socket API (718)

Kernel 1   705

Memory K1
706

Device
Driver
716

Device
Driver
717

Socket API (719)

Kernel 2   707

Memory K2
708

I/O Adapter
(720)

721   722
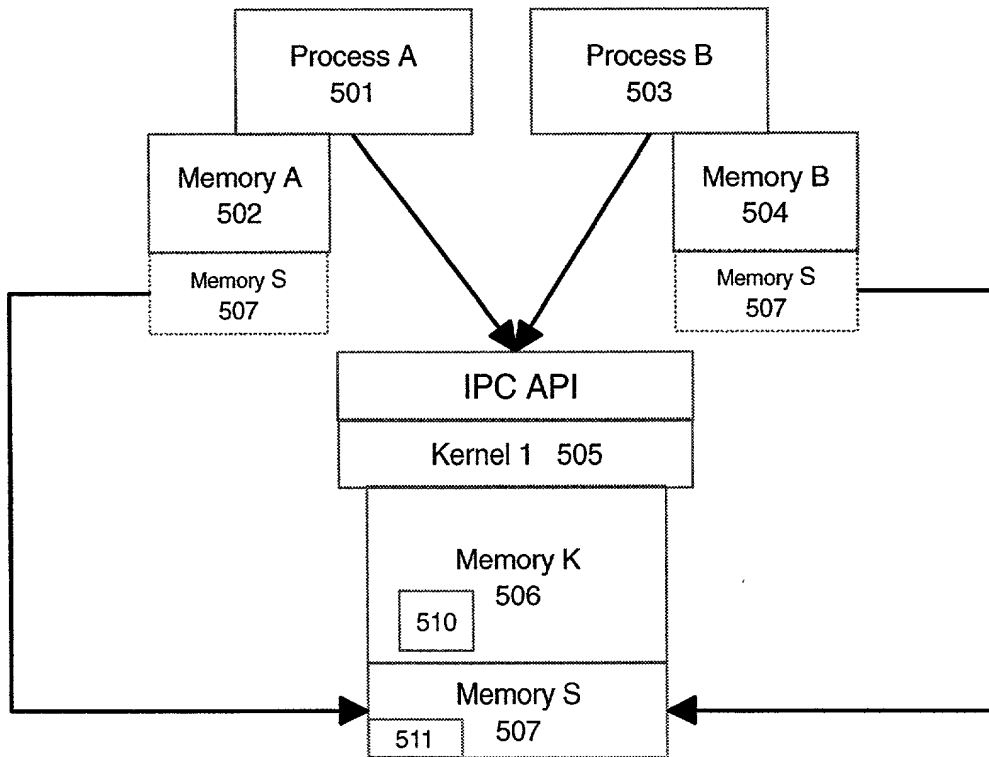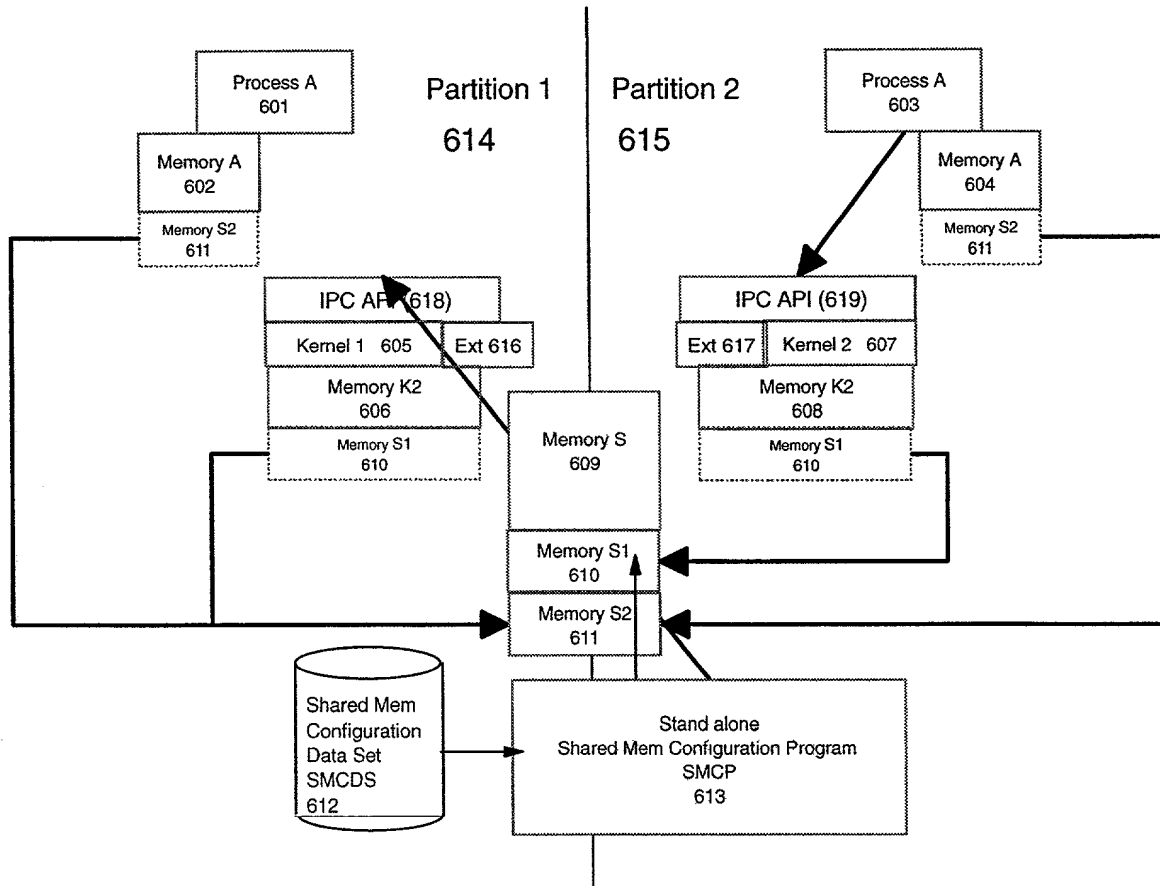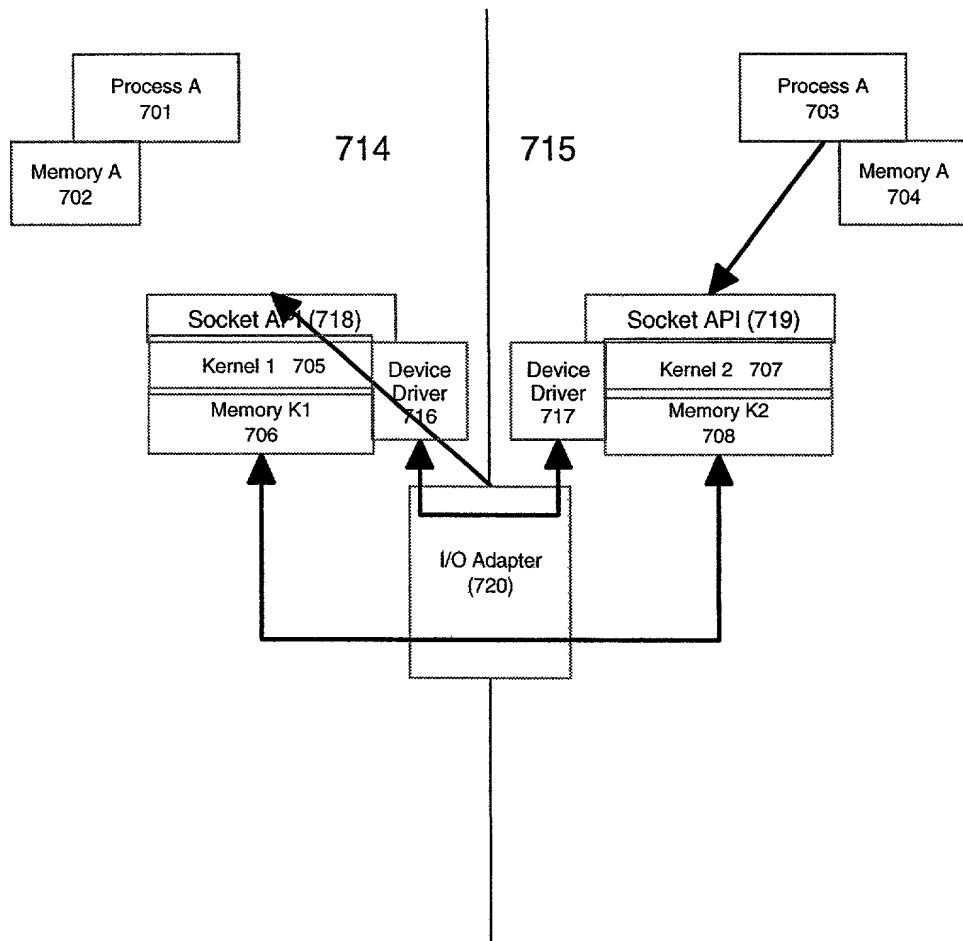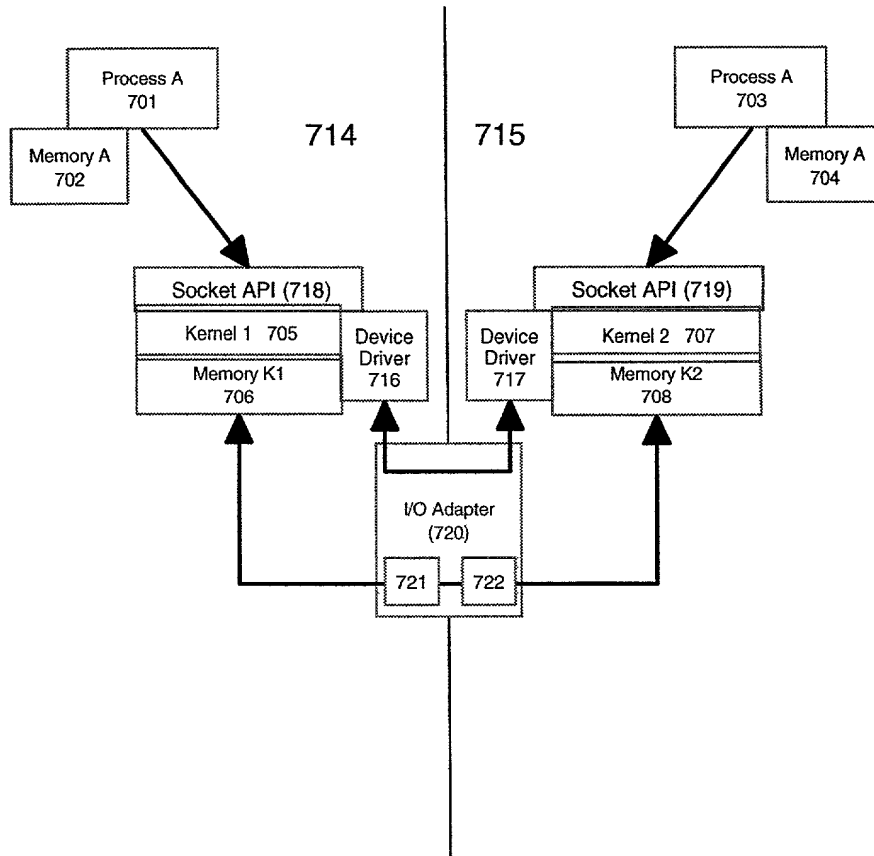
The Prior Art
Fig 7B

Fig 8

Fig 9

Data From Memory

```
Source        901
```

Byte
Alignment
902

Destination
903

Aligned
Data
904

Data to
Memory

Source
Address
Counter
905

Destination
Address
Counter
906

Select   907

Memory Address

Byte Counter
and State
Control
908

Fig 10

1000

MVXL	Count	Destinatio n	Source

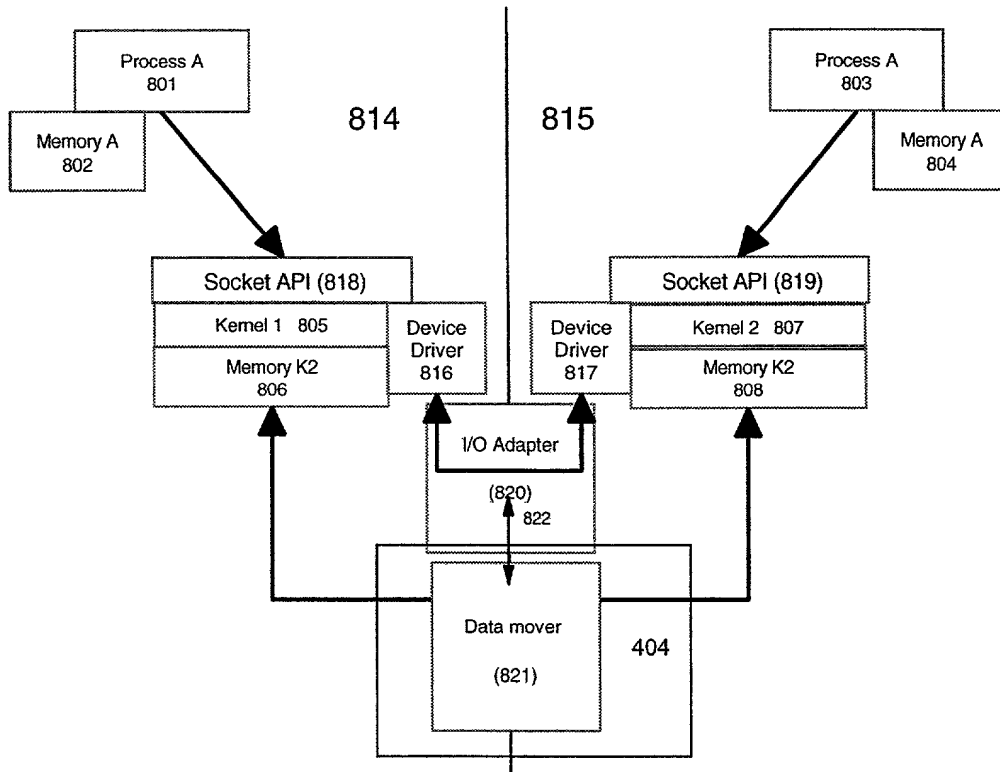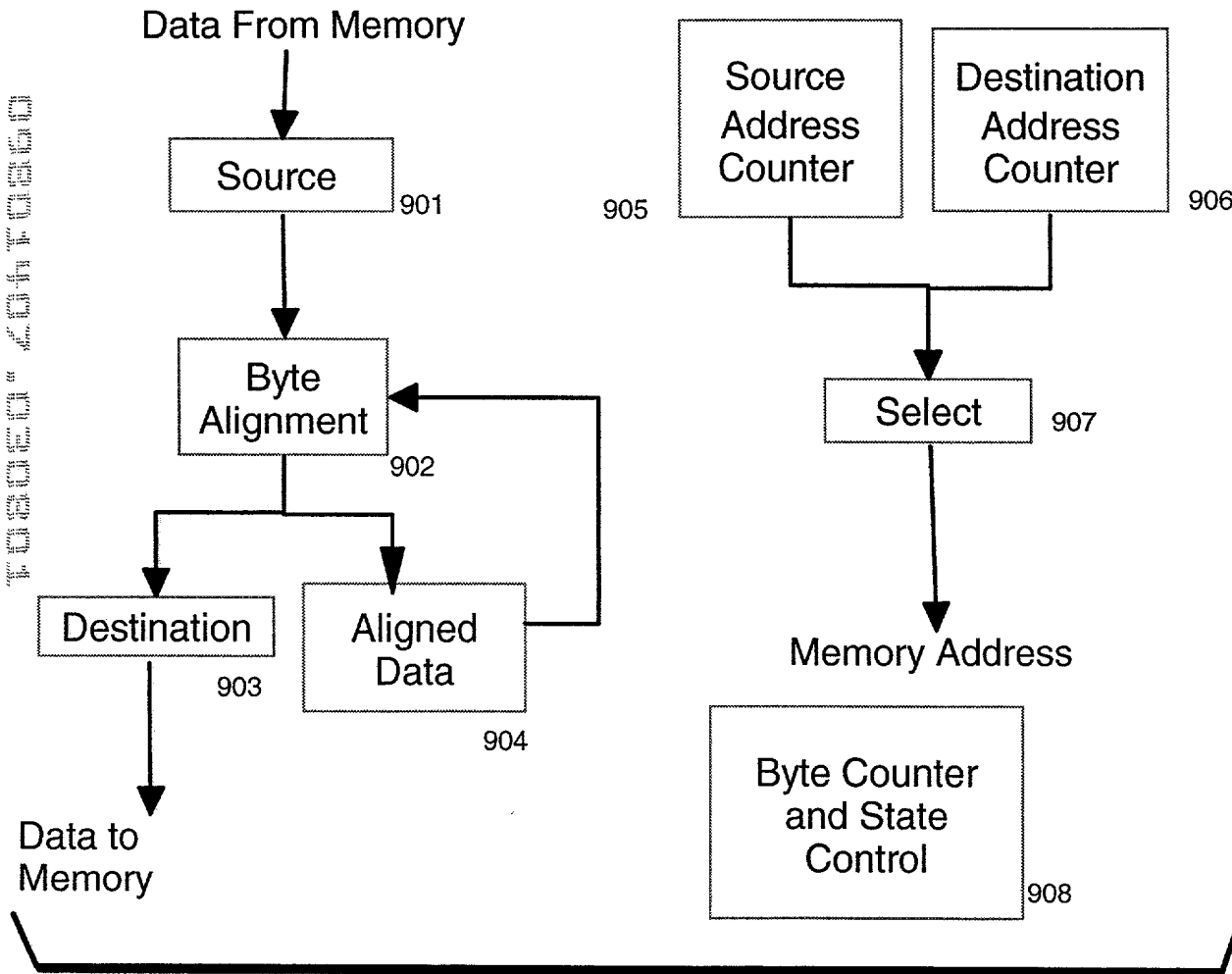MVXL moves the number of bytes specified by the count register from the physical address specified by the source register to the physical address specified by the destination register. The instruction is priveleged.

(MVCL performs the same function between virtual addressses.) Here the Device Driver loads the register with physical rather than virtual addresses allowing cross partition data movement.

Fig 11

1101. User calls Device Driver
    — Supplies
        Source Network ID
        Source Offset
        Destination Network ID
1102. Device driver transfers addresses to Adapter
1103. Adapter Translates Addresses
    — Looks up Phyical Base addresses from ID's (Table Lookup)
    — Obtains Lock and current Destination Offset
    — Adds offsets
    — Checks bounds
1104. Adapter loads count and addresses in registers
1105. Adapter executes Data Move
1106. Adapter Frees Lock
1107. Adapter notifies device Driver which "Returns" to user

Fig 12

1201. User calls Device Driver
 — Supplies
 Source Network ID
 Source Offset
 Destination Network ID
1202. Device driver sends addresses to adapter
1203. Adapter Translates
 — Looks up Phyical Base addresses from ID's (Table Lookup)
 — Obtains Lock and current Destination Offset*
 — Adds offsets
 — Checks bounds
 — Returns Lock and Physical addresses to Device Driver
1204. Device Driver executes Data Move
1205. Device Driver Frees Lock
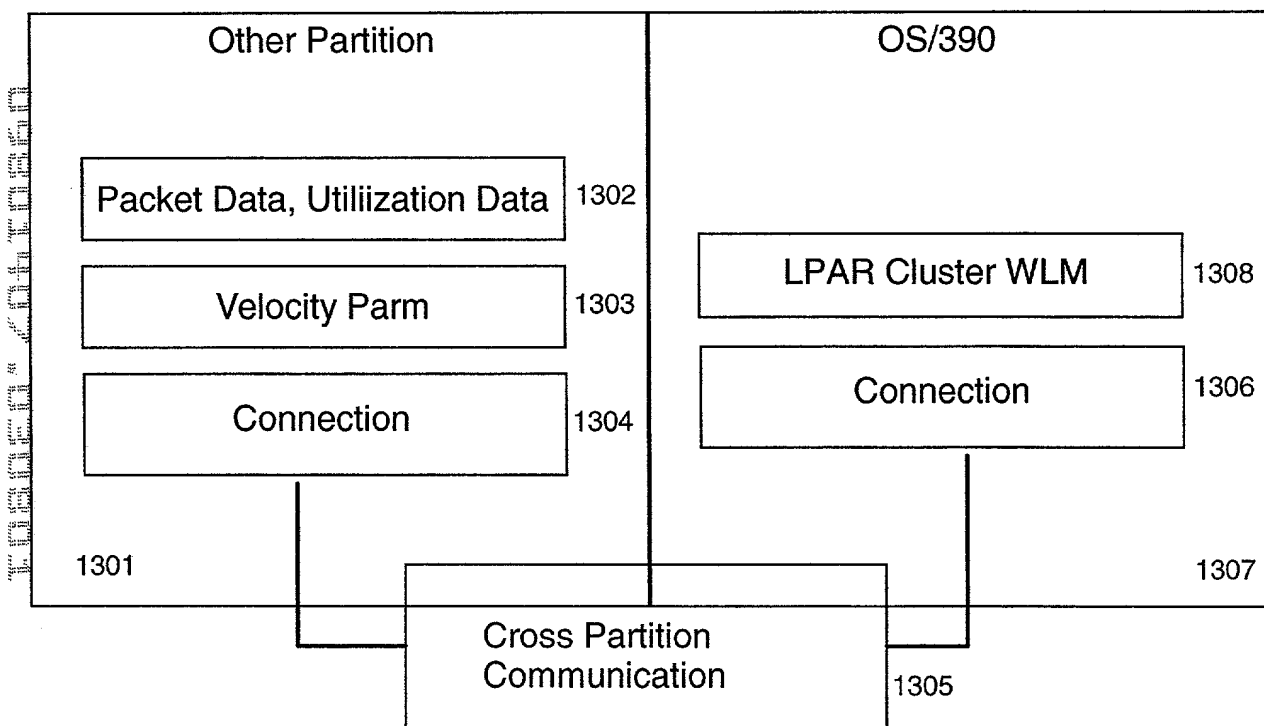1206. Device Driver Returns

Fig 13

Other Partition	OS/390

Packet Data, Utiliization Data	1302

Velocity Parm	1303

Connection	1304

LPAR Cluster WLM	1308

Connection	1306

1301

1307

Cross Partition Communication	

1305

Fig 14

Cum Packets v CPU

S = dT/dC

1405

R-square = 0.989   # pts = 43
1402
y = 0.802 + 1.13x

Fig 15

Device Driver A
1501

Memory A
1502

Memory S2
1511

1514

1515

Client Process B
1503

Memory B
1504

Memory S2
1511

IPC API

Kernel 1
1505

Ext 1516 Client

Memory K
1506

Memory S1
1510

IPC API

Ext 1517 Client

Kernel 2
1507

Memory K
1508

Memory S1
1510

Shared Storage
1509

Memory S1
1510

Work Queue
1513

Memory S2
1511

Return Data Buffer
1513

Fig 16